

HISTORIAS CLINCIAS VETERINARIAS

Trabajo final de Paz María Grecht Lort para el curso de Python - Nivel Inicial en el Centro de e-learning de la Facultad Regional Buenos Aires de la Universidad Tecnológica Nacional.

El objetivo de este trabajo final es desarrollar una aplicación que integre los conceptos aprendidos en cada unidad, permitiendo al alumno poner en práctica los conocimientos adquiridos y pasar de la teoría a la práctica.

Se solicita que la aplicación cumpla con los siguientes requisitos para su evaluación:

- Utilización de funciones.
- Uso de condicionales.
- Implementación de bucles.
- Integración de una base de datos SQLite3.
- Operaciones de alta, baja, consulta y modificación en la base de datos.
- Uso de expresiones regulares (regex).
- Notificación de eventos realizados.
- Descripción de la aplicación y su funcionamiento en un documento de Word.

Descripción

La aplicación que he desarrollado como trabajo final del curso de Python tiene como objetivo principal crear un registro de historias clínicas veterinarias. Con esta aplicación, se pueden ingresar y almacenar los siguientes campos de información:

- Datos de la mascota
 - Nombre
 - Edad
 - Color
 - Especie
 - Raza
 - Sexo
- Datos del dueño
 - Nombre del dueño
 - Correo electrónico del dueño
 - Teléfono del dueño
 - Dirección del dueño
 - Ciudad del dueño

Una vez que se ingresan estos datos, se pueden visualizar en un treeview, que es una estructura de datos que muestra los registros de forma jerárquica y organizada. Además, la aplicación cuenta con botones que permiten realizar diferentes acciones:

- Botón de alta: permite agregar un nuevo registro al registro de historias clínicas.
- Botón de selección: permite seleccionar un registro específico para ver sus detalles y poder realizar las modificaciones necesarias.
- Botón de modificación: permite editar y actualizar la información de un registro existente.
- Botón de baja: permite eliminar un registro del registro de historias clínicas.

Con esta aplicación, se facilita el manejo y organización de las historias clínicas veterinarias, brindando una forma eficiente de almacenar y visualizar la información.

Funcionamiento

La aplicación que he desarrollado para el trabajo final permite realizar diferentes acciones en un registro de historias clínicas veterinarias. A continuación, se describe el funcionamiento de cada una de las funciones implementadas:

1. Función "create_db": Esta función crea y devuelve una conexión a la base de datos SQLite llamada "db_tp_final.db".
2. Función "crear_tabla" : Esta función crea una tabla llamada "historiaclinica" en la base de datos si no existe. La tabla tiene columnas para almacenar los campos de una historia clínica veterinaria, como nombre de la mascota, edad, color, especie, raza, sexo, nombre del dueño, correo electrónico, teléfono, dirección y ciudad.
3. Función "update_tree":
 - a. Esta función actualiza el treeview con los registros de historias clínicas almacenados en la base de datos.
 - b. Elimina todos los registros existentes en el treeview.
 - c. Luego recupera los registros de la base de datos y los inserta en el treeview para mostrarlos.
4. Función "alta":
 - a. Esta función se encarga de agregar un nuevo registro al registro de historias clínicas.
 - b. Verifica que el correo electrónico ingresado sea válido utilizando expresiones regulares.
 - c. Si el correo electrónico es válido, se establece una conexión con la base de datos y se insertan los datos ingresados en la tabla "historiaclinica".
 - d. Luego, se actualiza el treeview para mostrar el nuevo registro agregado.

- e. Por último, se borran los campos de entrada de datos para permitir el ingreso de un nuevo registro.

5. Función "seleccionar":

- a. Esta función permite seleccionar un registro específico en el treeview para ver sus detalles.
- b. Se obtiene el ID del registro seleccionado en el treeview.
- c. Se establece una conexión con la base de datos y se realiza una consulta para obtener los datos correspondientes al ID seleccionado.
- d. Los datos obtenidos se insertan en los campos de entrada de datos para su visualización.

6. Función "modificar":

- a. Esta función permite modificar los datos de un registro existente en el registro de historias clínicas.
- b. Se obtiene el ID del registro seleccionado en el treeview.
- c. Se establece una conexión con la base de datos y se realiza una actualización de los datos correspondientes al ID seleccionado.
- d. Luego, se actualiza el treeview para reflejar los cambios realizados.
- e. Por último, se borran los campos de entrada de datos para permitir nuevas modificaciones.

7. Función "baja":

- a. Esta función permite eliminar un registro del registro de historias clínicas.
- b. Se obtiene el ID del registro seleccionado en el treeview.
- c. Se establece una conexión con la base de datos y se realiza la eliminación del registro correspondiente al ID seleccionado.

- d. Luego, se elimina el registro del treeview.
- 8. Función “vaciar_campos” : Esta función se utiliza para borrar todos los campos de entrada, dejándolos en blanco.
- 9. Función “cerrar”: Esta función se utiliza para cerrar la aplicación.

Estas funciones trabajan en conjunto para proporcionar una funcionalidad completa de gestión de historias clínicas veterinarias en la aplicación.

Pre-requisitos

Para poder utilizar la aplicación, es necesario tener instalado alguna de las versiones de Python3 y abrirla en un editor de código, como Visual Studio Code, y ejecutarla desde allí.

Si no tienes instalado Python3 en tu sistema operativo, puedes descargarlo desde el siguiente [enlace](#)

En caso de no tener ningún editor de código en tu computadora, recomiendo usar Visual Studio Code debido a sus numerosas extensiones y su interfaz de usuario intuitiva. Puedes descargar Visual Studio Code desde el siguiente enlace: <https://code.visualstudio.com/Download>

Construido con

- Python 3 - lenguaje de programación
- Tkinter - la interfaz de usuario del proyecto
- Sqlite3 - almacenar y gestionar los datos
- Regex - validar la entrada del usuario

Autora

con  por [Paz María Grecht Lort](#) 